AMENDMENTS TO THE CLAIMS:

1. (Original) A method of performing a computation in a programmable processor, the programmable processor having a first memory system having a first data path width, and a second memory system and a third memory system each of the second memory system and the third memory system having a data path width which is greater than the first data path width, the method comprising the steps of:

copying a first memory operand portion from the first memory system to the second memory system, the first memory operand portion having the first data path width;

copying a second memory operand portion from the first memory system to the second memory system, the second memory operand portion having the first data path width and being catenated in the second memory system with the first memory operand portion, thereby forming first catenated data;

copying a third memory operand portion from the first memory system to the third memory system, the third memory operand portion having the first data path width;

copying a fourth memory operand portion from the first memory system to the third memory system, the fourth memory operand portion having the first data path width and being catenated in the third memory system with the third memory operand portion, thereby forming second catenated data; and

performing a computation of a single instruction using the first catenated data and the second catenated data.

2. (Original) The method of claim 1 wherein the step of performing a computation further comprises reading a portion of the first catenated data and a portion of the second catenated data each of which is greater in width than the first data path width and using the

portion of the first catenated data and the portion of the second catenated data to perform the computation.

3. (Original) The method of claim 2 further comprising the step of specifying a memory address of each of the first catenated data and of the second catenated data within the first memory system.

4. (Original) The method of claim 3 further comprising the step of specifying a memory operand size and a memory operand shape of each of the first catenated data and the second catenated data.

5. (Currently amended) The method of claim 2 further comprising the step of checking ~~the~~ validity of each of the first catenated data in the second memory system and the second catenated data in the third memory system, and, if valid, permitting a subsequent instruction to use the first and second catenated data without copying from the first memory system.

6. (Original) The method of claim 2 wherein the step of performing a computation further comprises performing a convolution of partitioned elements contained in the first catenated data with partitioned elements contained in the second catenated data, forming a convolution data, extracting a specified subfield of the convolution data and catenating extracted data, and forming a catenated result having a size equal to that of the functional unit data path width.

7. (Original) The method of claim 2, wherein the step of performing a computation further comprises performing a transform of partitioned elements contained in the first catenated

data using coefficients contained in the second catenated data, thereby forming a transform data, extracting a specified subfield of the transform data, thereby forming an extracted data and catenating the extracted data.

8. (Original) A method of performing a computation in a programmable processor, the programmable processor having a first memory system having a first data path width, and a second and a third memory system having a data path width which is greater than the first data path width, the method comprising the steps of:

copying a first memory operand portion from the first memory system to the second memory system, the first memory operand portion having the first data path width;

copying a second memory operand portion from the first memory system to the second memory system, the second memory operand portion having the first data path width and being catenated in the second memory system with the first memory operand portion, thereby forming first catenated data;.

performing a computation of a single instruction using the first catenated data and producing a second catenated data,

copying a third memory operand portion from the third memory system to the first memory system, the third memory operand portion having the first data path width and containing a portion of the second catenated data; and

copying a fourth memory operand portion from the third memory system to the first memory system, the fourth memory operand portion having the first data path width and containing a portion of the second catenated data, wherein the fourth memory operand portion is catenated in the third memory system with the third memory operand portion.

9. (Original) The method of claim 8 wherein the step of performing a computation further comprises the step of reading a portion of the first catenated data which is greater in width than the first data path width and using the portion of the first catenated data to perform the computation.

10. (Original) The method of claim 9 further comprising the step of specifying a memory address of each of the first catenated data and of the second catenated data within the first memory system.

11. (Original) The method of claim 10 further comprising the step of specifying a memory operand size and a memory operand shape of each of the first catenated data and the second catenated data.

12. (Currently amended) The method of claim 9 further comprising the step of checking ~~the~~ validity of the first catenated data in the second memory system, and, if valid, permitting a subsequent instruction to use the first catenated data without copying from the first memory system.

13. (Original) The method of claim 8 wherein the step of performing a computation further comprises the step of performing a transform of partitioned elements contained in the first catenated data, thereby forming a transform data, extracting a specified subfield of the transform data, thereby forming an extracted data and catenating the extracted data, forming the second catenated data.

14. (Currently amended) The method of claim 13 ~~claim 8~~ wherein the step of performing a computation further comprises the step of combining using Boolean arithmetic a

portion of the extracted data with an accumulated Boolean data, combining partitioned elements of the accumulated Boolean data using Boolean arithmetic, forming combined Boolean data, determining a ~~the~~ most significant bit of the extracted data from the combined Boolean data, and returning a result comprising a ~~the~~ position of the most significant bit to a register.

15. (Currently amended) The method of claim 8 further comprising manipulating a first and a second validity information corresponding to first and second catenated data, wherein after completion of an instruction specifying a memory address of first catenated data, ~~the~~ contents of second catenated data are provided to the first memory system in place of first catenated data.

16. (Original) A programmable processor comprising:

a first memory system having a first data path width;

a second memory system and a third memory system, wherein each of the second memory system and the third memory system have a data path width which is greater than the first data path width;

a first copying module configured to copy a first memory operand portion from the first memory system to the second memory system, the first memory operand portion having the first data path width, and configured to copy a second memory operand portion from the first memory system to the second memory system, the second memory operand portion having the first data path width and being catenated in the second memory system with the first memory operand portion, thereby forming first catenated data;

a second copying module configured to copy a third memory operand portion from the first memory system to the third memory system, the third memory operand portion having the

first data path width, and configured to copy a fourth memory operand portion from the first memory system to the third memory system, the fourth memory operand portion having the first data path width and being catenated in the third memory system with the third memory operand portion, thereby forming second catenated data; and

a functional unit configured to perform computations using the first catenated data and the second catenated data.

17. (Original) The programmable processor of claim 16 wherein the functional unit is further configured to read a portion of each of the first catenated data and the second catenated data which is greater in width than the first data path width and use the portion of each of the first catenated data and the second catenated data to perform the computation.

18. (Original) The programmable processor of claim 17 wherein the functional unit is further configured to specify a memory address of each of the first catenated data and of the second catenated data within the first memory system.

19. (Original) The programmable processor of claim 18 wherein the functional unit is further configured to specify a memory operand size and a memory operand shape of each of the first catenated data and the second catenated data.

20. (Currently amended) The programmable processor of claim 17 further comprising a control unit configured to check ~~the~~ validity of each of the first catenated data in the second memory system and the second catenated data in the third memory system, and, if valid, permitting a subsequent instruction to use each of the first catenated data and the second catenated data without copying from the first memory system.

21. (Original) The programmable processor of claim 16, wherein the functional unit is further configured to convolve partitioned elements contained in the first catenated data with partitioned elements contained in the second catenated data, forming a convolution data, extract a specified subfield of the convolution data and catenate extracted data, forming a catenated result having a size equal to that of the functional unit data path width.

22. (Original) The programmable processor of claim 16 wherein the functional unit is further configured to perform a transform of partitioned elements contained in the first catenated data using coefficients contained in the second catenated data, thereby forming a transform data, extract a specified subfield of the transform data, thereby forming an extracted data and catenate the extracted data.

23. (Original) A programmable processor comprising:

a first memory system having a first data path width,

a second memory system and a third memory system each of the second memory system and the third memory system having a data path width which is greater than the first data path width;

a first copying module configured to copy a first memory operand portion from the first memory system to the second memory system, the first memory operand portion having the first data path width, and configured to copy a second memory operand portion from the first memory system to the second memory system, the second memory operand portion having the first data path width and being catenated in the second memory system with the first memory operand portion, thereby forming first catenated data; a second copying module configured to copy a third memory operand portion from the third memory system to the first memory system, the

third memory operand portion having the first data path width and containing a portion of a second catenated data, and copy a fourth memory operand portion from the third memory system to the first memory system, the fourth memory operand portion having the first data path width and containing a portion of the second catenated data, wherein the fourth memory operand portion is catenated in the third memory system with the third memory operand portion; and

a functional unit configured to perform computations using the first catenated data and the second catenated data.

24. (Original) The programmable processor of claim 23 wherein the functional unit is further configured to read a portion of the first catenated data which is greater in width than the first data path width and use the portion of the first catenated data to perform the computation.

25. (Original) The programmable processor of claim 24 wherein the functional unit is further configured to specify a memory address of each of the first catenated data and of the second catenated data within the first memory system.

26. (Original) The programmable processor of claim 25 wherein the functional unit is further configured to specify a memory operand size and a memory operand shape of each of the first catenated data and the second catenated data.

27. (Currently amended) The programmable processor of claim 24 further comprising a control unit configured to check ~~the~~ validity of the first catenated data in the second memory system, and, if valid, permitting a subsequent instruction to use the first catenated data without copying from the first memory system.

28. (Original) The programmable processor of claim 23 wherein the functional unit is further configured to transform partitioned elements contained in the first catenated data, thereby forming a transform data, extract a specified subfield of the transform data, thereby forming an extracted data and catenate the extracted data, forming the second catenated data.

29. (Currently amended) The programmable processor of claim 28 ~~claim 23~~ wherein the functional unit is further configured to combine using Boolean arithmetic a portion of the extracted data with an accumulated Boolean data, combine partitioned elements of the accumulated Boolean data using Boolean arithmetic, forming combined Boolean data, determine a ~~the~~ most significant bit of the extracted data from the combined Boolean data, and provide a result comprising a ~~the~~ position of the most significant bit.

30. (Currently amended) The programmable processor of claim 23 further comprising a control unit configured to manipulate a first and a second validity information corresponding to first and second catenated data, wherein after completion of an instruction specifying a memory address of first catenated data, ~~the~~ contents of second catenated data are provided to the first memory system in place of first catenated data.